

# Quadratic Bézier Curve Approximation By Circular Arcs Within A Tolerance Band

Seyed Amir Hossein Siahposhha

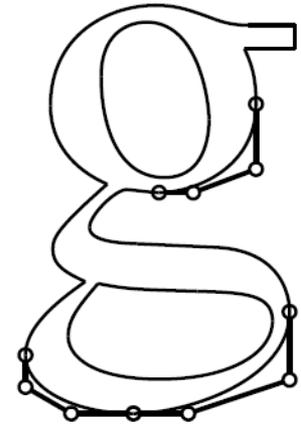
*Seminar aus Informatik  
Department of Computer Sciences*

# Outline

- Motivation
- Bézier Curve
- The Bisection (de-Casteljau) Algorithm
- Previous Approximation Methods and Challenges
- Hausdorff Distance
- Quadratic Bézier-Like Algorithm
- New Approximation Methods

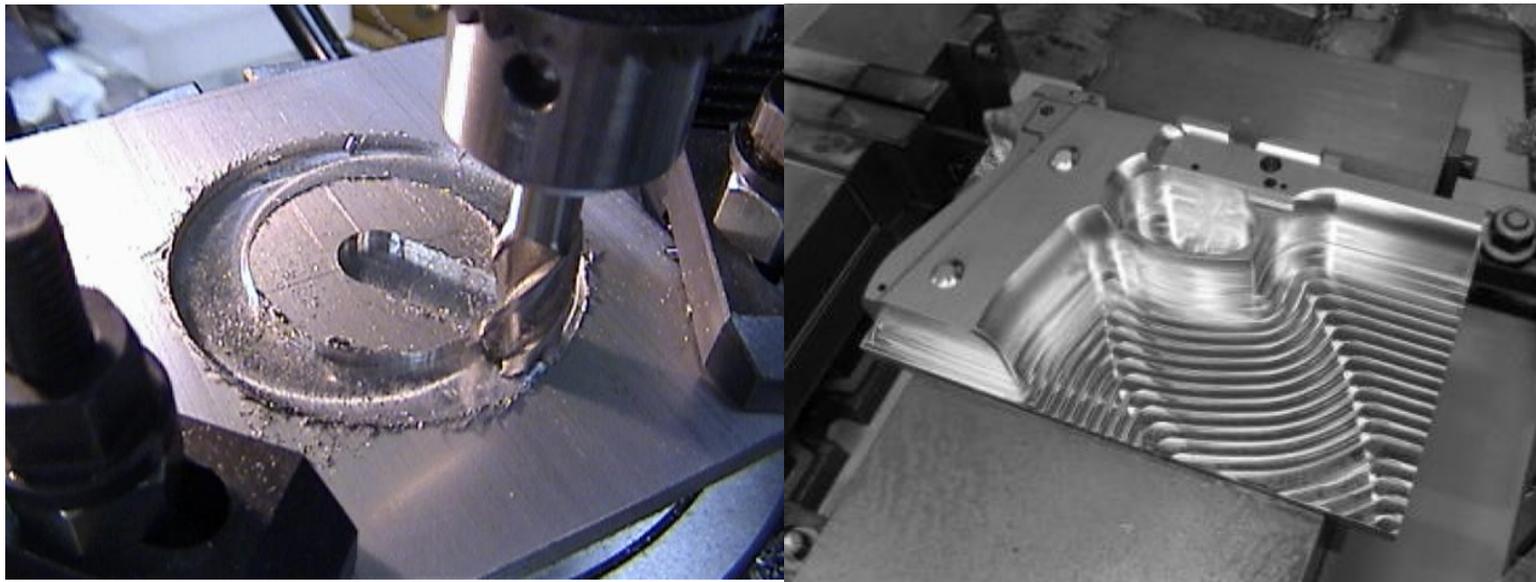
# Motivation

- Bézier curve is a classical type of general curve in CAD/CAM, which is widely used in various interesting CAD/CAM and Computer Graphics applications.
- The tool path for CNC (Computer Numerical Control) machinery can use piecewise curves made from circular arc and straight line segments.
- In between, designed shapes consisting of tangential circular arc segments are better and easier to use than the polygons consisting of line segments as the tool path of CNC machinery.
- The combination of these two topics is the main base of Bézier curve approximation by tangential circular arcs.



# CNC Machining

- In programming the tool path of CNC machinery, a smaller number of arc segments in approximation can reduce the number of instructions and tool motions.
- Therefore, to improve production efficiency we need to approximate Bézier curves by tangential circular arc segments with fewer arc segments that are as small as possible.



# Bézier Curve

- The pioneers of Bézier Curve were Pierre Bézier who worked as an engineer at Renault, and Paul de-Casteljau who worked at Citroen; This reflects the usability and importance of these classic curves in industry and engineering.
- General curves can be created by any type of algebraic formula but applied mathematics is not able to do every processing on every formula.
- Because of this huge limitation, just the simple curves that is easy to process are using in engineering and industry.

# Bézier Curve

Every Bézier curve can represent a Bézier curve of higher degree.

➤ Linear Bézier Curve

$$\mathbf{B}(t) = \mathbf{P}_0 + t(\mathbf{P}_1 - \mathbf{P}_0) = (1-t)\mathbf{P}_0 + t\mathbf{P}_1, \quad t \in [0, 1]$$

➤ Quadratic Bézier Curve

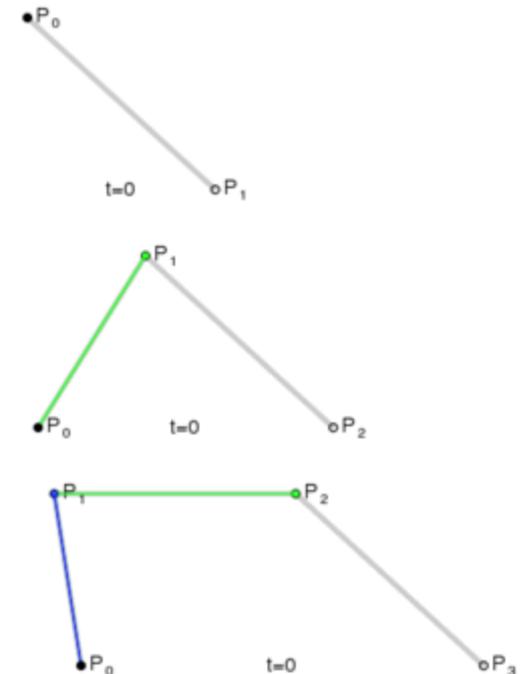
$$\mathbf{B}(t) = (1-t)[(1-t)\mathbf{P}_0 + t\mathbf{P}_1] + t[(1-t)\mathbf{P}_1 + t\mathbf{P}_2], \quad t \in [0, 1]$$

$$\mathbf{B}(t) = (1-t)^2\mathbf{P}_0 + 2(1-t)t\mathbf{P}_1 + t^2\mathbf{P}_2, \quad t \in [0, 1].$$

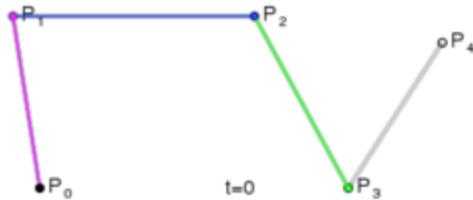
➤ Cubic Bézier Curves

$$\mathbf{B}(t) = (1-t)\mathbf{B}_{\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2}(t) + t\mathbf{B}_{\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3}(t), \quad t \in [0, 1].$$

$$\mathbf{B}(t) = (1-t)^3\mathbf{P}_0 + 3(1-t)^2t\mathbf{P}_1 + 3(1-t)t^2\mathbf{P}_2 + t^3\mathbf{P}_3, \quad t \in [0, 1].$$



# Bézier Curve Formulation



Quartic Bézier Curve



Fifth-order Bézier Curve

$$\mathbf{B}_{P_0P_1P_2P_3P_4P_5}(t) = \mathbf{B}(t) = (1-t)^5\mathbf{P}_0 + 5t(1-t)^4\mathbf{P}_1 + 10t^2(1-t)^3\mathbf{P}_2 + 10t^3(1-t)^2\mathbf{P}_3 + 5t^4(1-t)\mathbf{P}_4 + t^5\mathbf{P}_5, \quad t \in [0, 1].$$

➤ General Recursive Formula

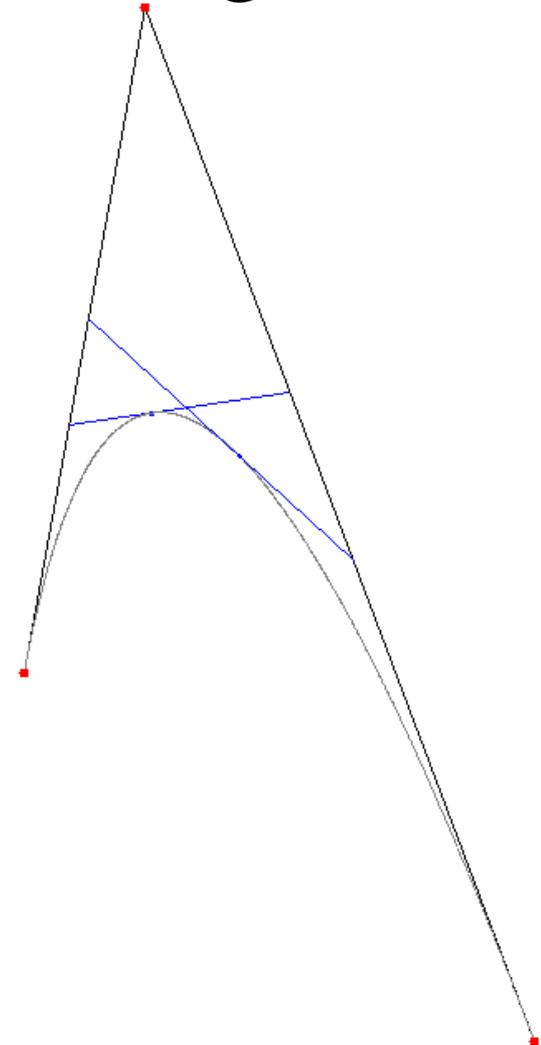
$$\mathbf{B}(t) = \mathbf{B}_{P_0P_1\dots P_n}(t) = (1-t)\mathbf{B}_{P_0P_1\dots P_{n-1}}(t) + t\mathbf{B}_{P_1P_2\dots P_n}(t)$$

➤ General Formula by Bernstein Polynomials

$$\begin{aligned} \mathbf{B}(t) &= \sum_{i=0}^n \binom{n}{i} (1-t)^{n-i} t^i \mathbf{P}_i \\ &= (1-t)^n \mathbf{P}_0 + \binom{n}{1} (1-t)^{n-1} t \mathbf{P}_1 + \dots \\ &\quad \dots + \binom{n}{n-1} (1-t) t^{n-1} \mathbf{P}_{n-1} + t^n \mathbf{P}_n, \quad t \in [0, 1], \end{aligned}$$

# Bézier Curve Construction Algorithm

- Every Bézier Curve is dividable to Bézier sub-curves with the same degree by intersection of tangents lines.
- This fractal features is the base of Bézier Curve Construction Algorithm .
- The Bisection (de-Casteljau) algorithm is known as the fundamental geometric construction algorithm.
- Almost methods for approximating Bézier curves are using the de-Casteljau algorithm.



# The Bisection (de-Casteljau) Algorithm

Quadratic Bezier Curve ( $P_0^0, P_0^1, P_0^2$ )

{

$$P_1^0 \leftarrow \frac{P_0^0 - P_0^1}{2} + P_0^1;$$

$$P_1^1 \leftarrow \frac{P_0^1 - P_0^2}{2} + P_0^2;$$

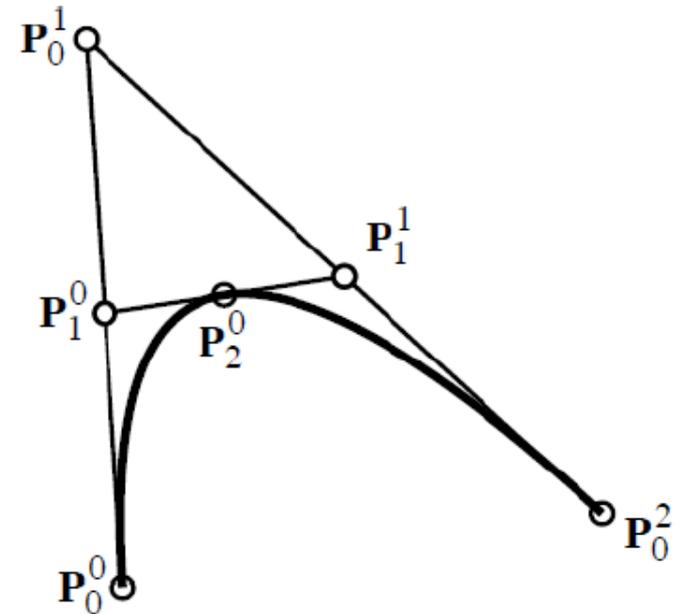
$$P_2^0 \leftarrow \frac{P_1^0 - P_1^1}{2} + P_1^1;$$

*Draw*( $P_2^0$ );

Quadratic Bezier Curve ( $P_0^0, P_1^0, P_2^0$ )

Quadratic Bezier Curve ( $P_2^0, P_1^1, P_0^2$ )

}



1 curve



2 curves



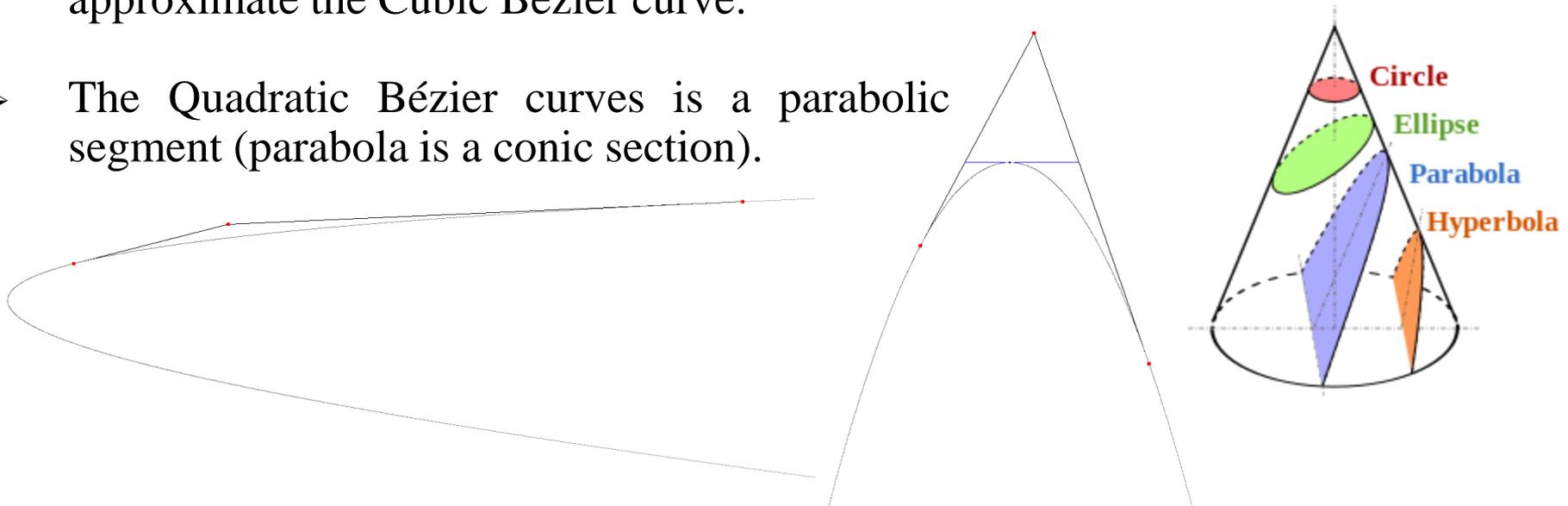
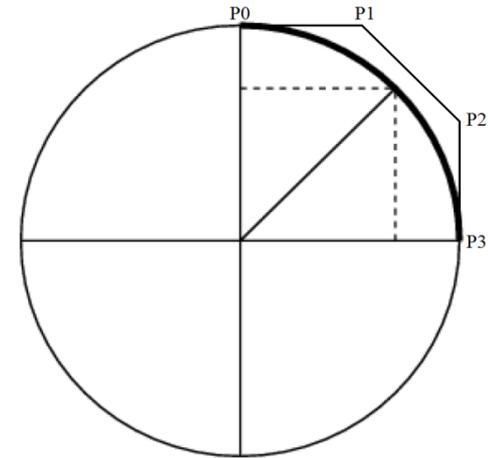
4 curves



8 curves

# Bézier Curve and Circular Arc

- The high degree Bézier Curves are too complex to process and approximation; therefore the Quadratic and Cubic Bézier curve are more common to use in CAD/CAM.
- Circular arc can be exactly represented by Cubic Bézier curve but not Quadratic one; therefore there are more different methods to approximate the Cubic Bézier curve.
- The Quadratic Bézier curves is a parabolic segment (parabola is a conic section).



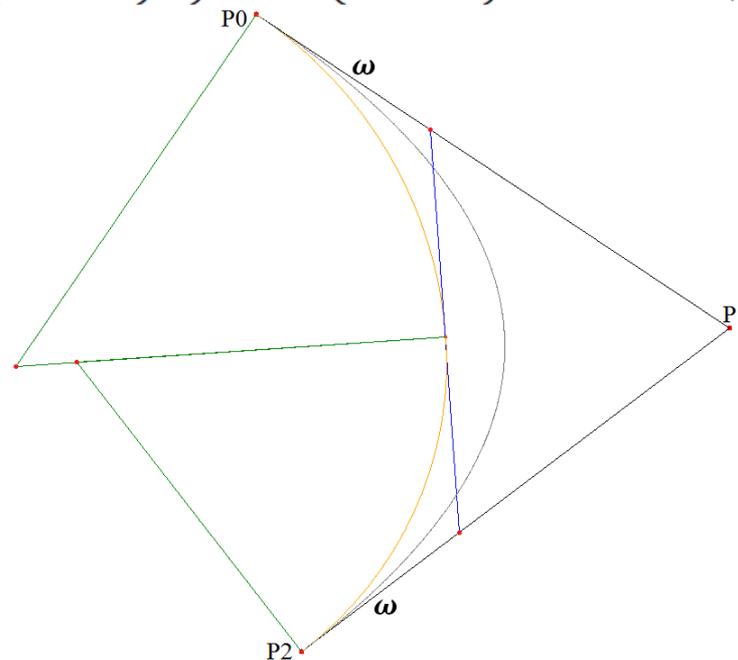
# Biarc Method

- Biarc method is the most important method to approximate the quadratic Bézier curves by circular arcs (that presented by Walton and Meek)
- A biarc made of two circular arcs, is associated with each quadratic Bézier curve segment to match both end points and the corresponding tangent vectors.

$$\mathbf{a} = \|\mathbf{P}_0 - \mathbf{P}_1\|, \quad \mathbf{b} = \|\mathbf{P}_1 - \mathbf{P}_2\|, \quad \mathbf{c} = \|\mathbf{P}_0 - \mathbf{P}_2\|, \quad \cos \theta = \frac{a^2 + b^2 - c^2}{2ab}$$

$$(\omega a + (1 - \omega)b)^2 = ((1 - \omega)a)^2 + (\omega b)^2 - 2(1 - \omega)\omega ab \cos \theta, \quad 0 \leq \omega \leq 1$$

$$\omega = \frac{\pm c(a+b) - c^2}{(a+b)^2 - c^2}, \quad 0 \leq \omega \leq 1$$



# Hausdorff Distance

- Hausdorff Distance is obtained from the maximum distance between two curves.
- The Hausdorff Distance is using in CAD/CAM Approximation Theories within a tolerance band.
- The general mathematic definition for Hausdorff Distance is:

$$d_H(p, q) = \max \left\{ \max_{s \in [a, b]} \min_{t \in [c, d]} |p(s) - q(t)|, \max_{t \in [c, d]} \min_{s \in [a, b]} |p(s) - q(t)| \right\}$$

# Hausdorff Distance in Biarc Method

- Walton and Meek has presented the formula with  $O(n)$  processing time to find the Hausdorff Distances of every quadratic Bézier curve segment  $Q(u)$  from its biarc  $B(Cl, Rl, Cr, Rr)$  measured along a radial direction of the biarc, where  $Cl$  and  $Rl$  are the centre and radius of the left arc, and  $Cr$  and  $Rr$  are the centre and radius of the right arc as follow:

$$d_H(Q, B) = \max_{0 \leq u \leq 1} \{ \rho_l(u), \rho_r(u) \}$$

$$\rho_l(u) = \left| R_l - \|Q(u) - C_l\| \right|, \quad 0 \leq u \leq 1$$

$$\rho_r(u) = \left| R_r - \|Q(u) - C_r\| \right|, \quad 0 \leq u \leq 1$$

# The Bisection Challenge

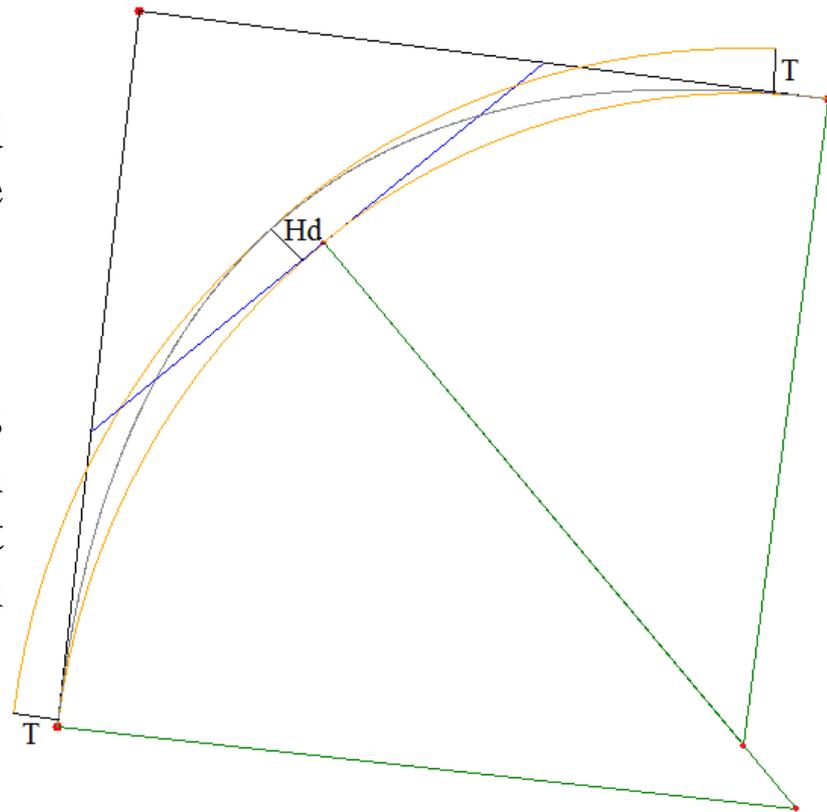
- In the Biarc method, the Hausdorff Distances can be smaller than or equal to Tolerance Band.

$$Hd \leq T$$

- For the minimal number of approximated circular arcs, the Hausdorff Distances must be just equal to Tolerance Band.

$$Hd = T$$

- Therefore, every method which follows Bisection algorithm to divide Bézier curve, can generate smaller Bézier curve segments, that needs more number of circular arcs than minimal approximation.

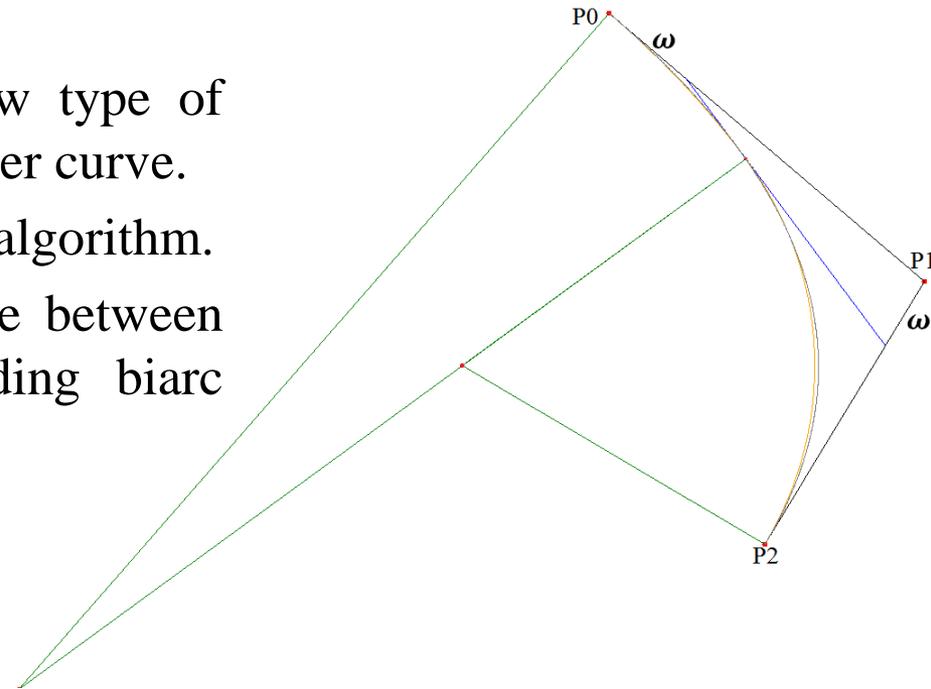


# Hausdorff Distance Challenge

- The second problem of the biarc method is the calculation of Hausdorff Distance.
- The processing time of this calculation is depend on the program accuracy and it can be a huge amount for the high accuracy.

# Quadratic Bézier-Like

- Quadratic Bézier-Like curve is a new type of curve, which is close to Quadratic Bézier curve.
- This curve is constructed by Bisection algorithm.
- The calculation of Hausdorff Distance between this curve segments and corresponding biarc needs just  $O(1)$  processing time.



$$a = \|P_0 - P_1\|, \quad b = \|P_1 - P_2\|, \quad c = \|P_0 - P_2\|, \quad \cos \theta = \frac{a^2 + b^2 - c^2}{2ab}$$

$$(\omega a + (1 - \omega)b)^2 = ((1 - \omega)a)^2 + (\omega b)^2 - 2(1 - \omega)\omega ab \cos \theta, \quad 0 \leq \omega \leq 1$$

$$\omega = \frac{-(a+b)(b-3a) - c^2 \pm \sqrt{((a+b)(b-3a) + c^2)^2 - 4((a+b)^2 - c^2)(a^2 - b^2)}}{2((a+b)^2 - c^2)}, \quad 0 \leq \omega \leq 1$$

*Quadratic Bezier-Like* ( $P_0^0, P_0^1, P_0^2$ )

{

*If* ( $\|P_0^0 - P_0^1\| = \|P_0^1 - P_0^2\|$ )

{

$$P_1^0 \leftarrow \frac{P_0^0 - P_0^1}{2} + P_0^1;$$

$$P_1^1 \leftarrow \frac{P_0^1 - P_0^2}{2} + P_0^2;$$

$$P_2^0 \leftarrow \frac{P_1^0 - P_1^1}{2} + P_1^1;$$

*Draw*( $P_2^0$ );

*Quadratic Bezier Like* ( $P_0^0, P_1^0, P_2^0$ )

*Quadratic Bezier Like* ( $P_2^0, P_1^1, P_0^2$ )

}

*Else*

{

$\omega \leftarrow$  *find the Bezier coefficient to divide* ( $P_0^0, P_0^1, P_0^2$ );

$$P_1^0 \leftarrow (P_0^0 - P_0^1)\omega + P_0^1;$$

$$P_1^1 \leftarrow (P_0^1 - P_0^2)\omega + P_0^2;$$

$P_2^0 \leftarrow$  *find the touch point of biarc on tangent line* ( $P_1^0, P_1^1$ );

*Draw*( $P_2^0$ )

*Quadratic Bezier Like* ( $P_0^0, P_1^0, P_2^0$ )

*Quadratic Bezier Like* ( $P_2^0, P_1^1, P_0^2$ )

}

}

$Ca(C, r) \leftarrow$  *find Circular Arc*( $P_0^0, P_0^1, P_0^2$ );

*If* ( $\|C - P_2^0\| < \textit{Tolerance band}$ )

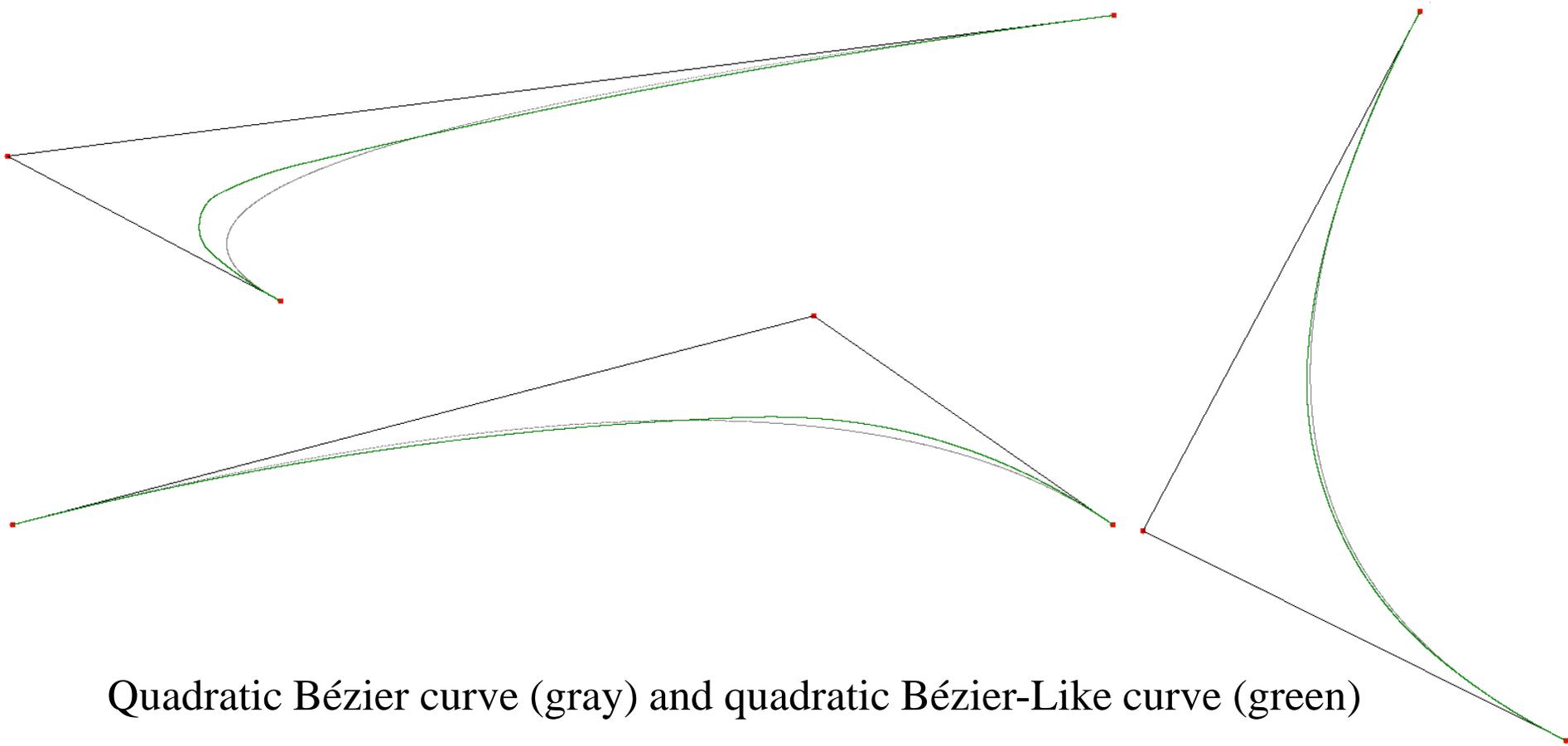
{

*Draw Circular Arc*( $Ca(C, r)$ );

*Return*;

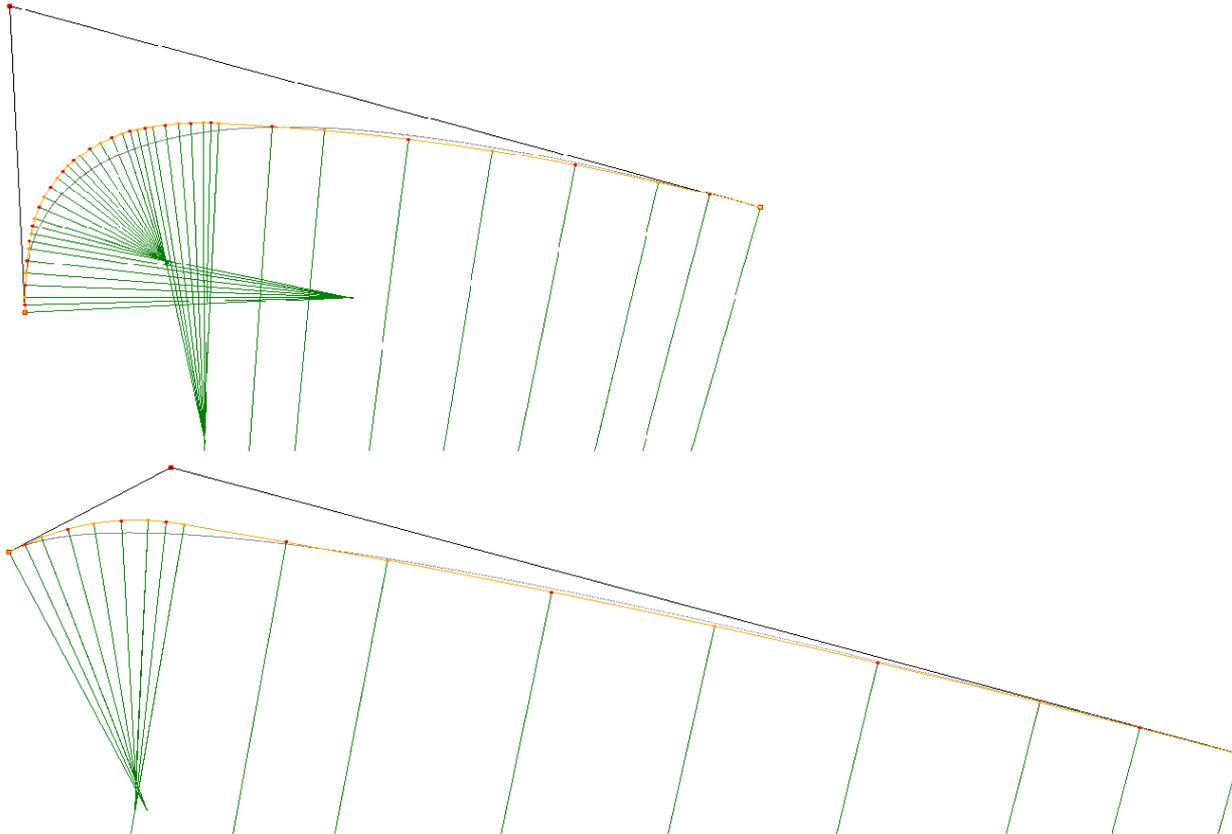
}

# The Algorithm Results



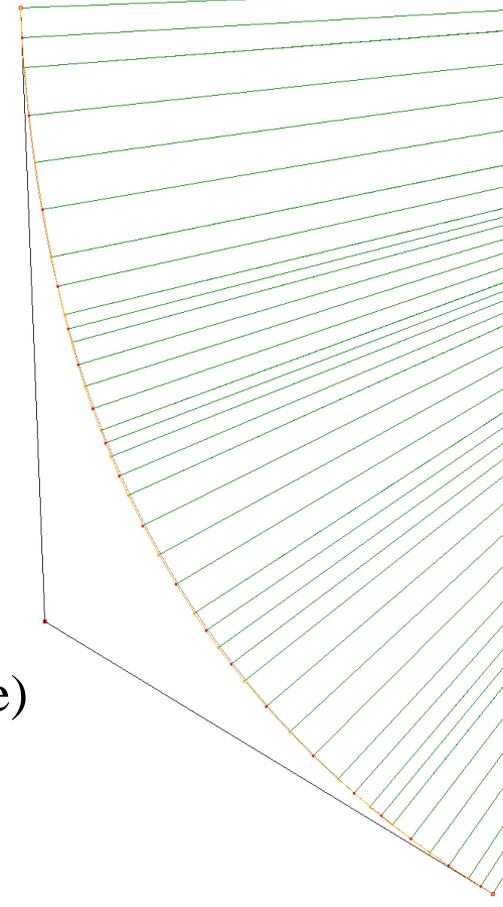
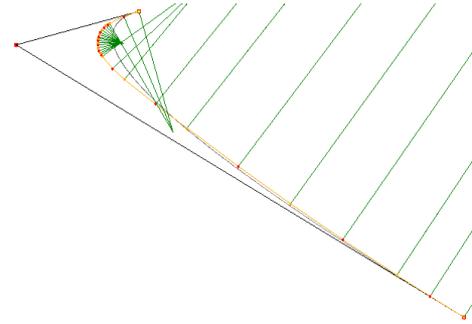
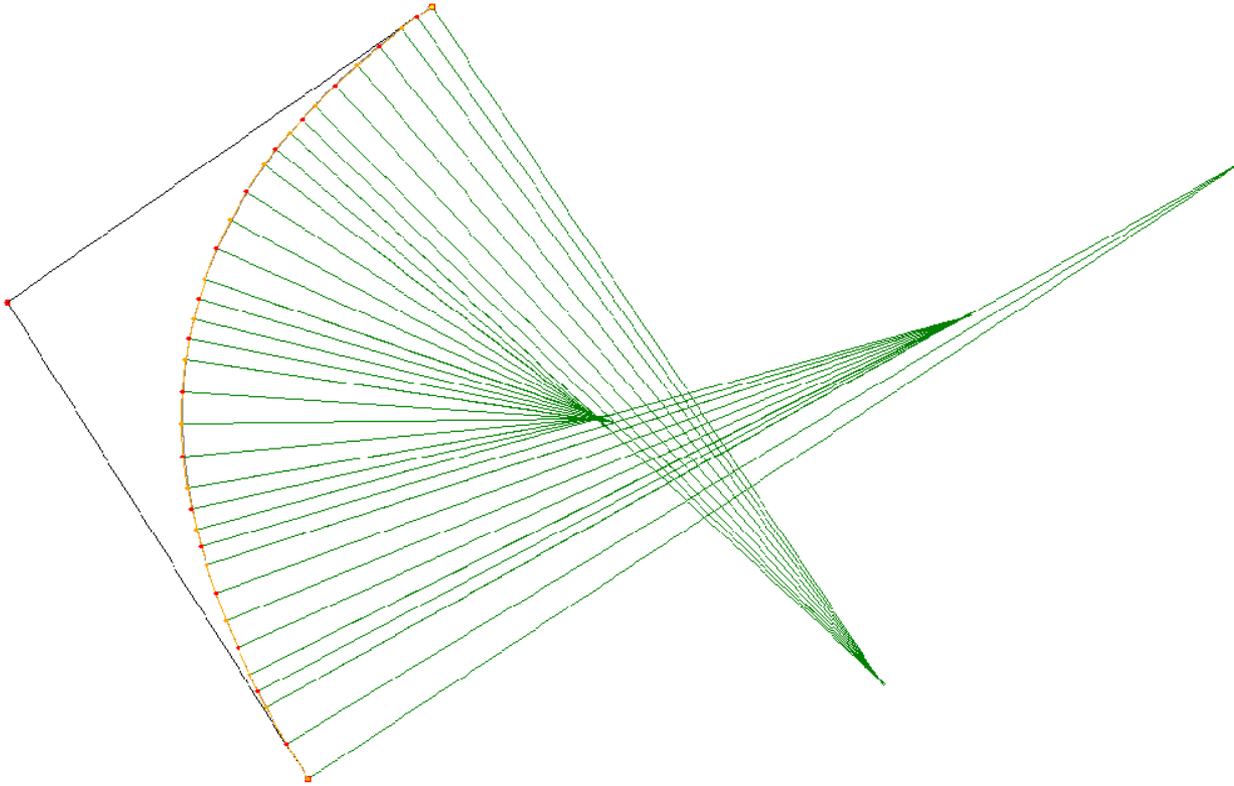
Quadratic Bézier curve (gray) and quadratic Bézier-Like curve (green)

# The Algorithm Results



Quadratic Bézier-like curve approximation by circular arc (orange)  
and corresponding quadratic Bézier curve (gray)

# The Algorithm Results



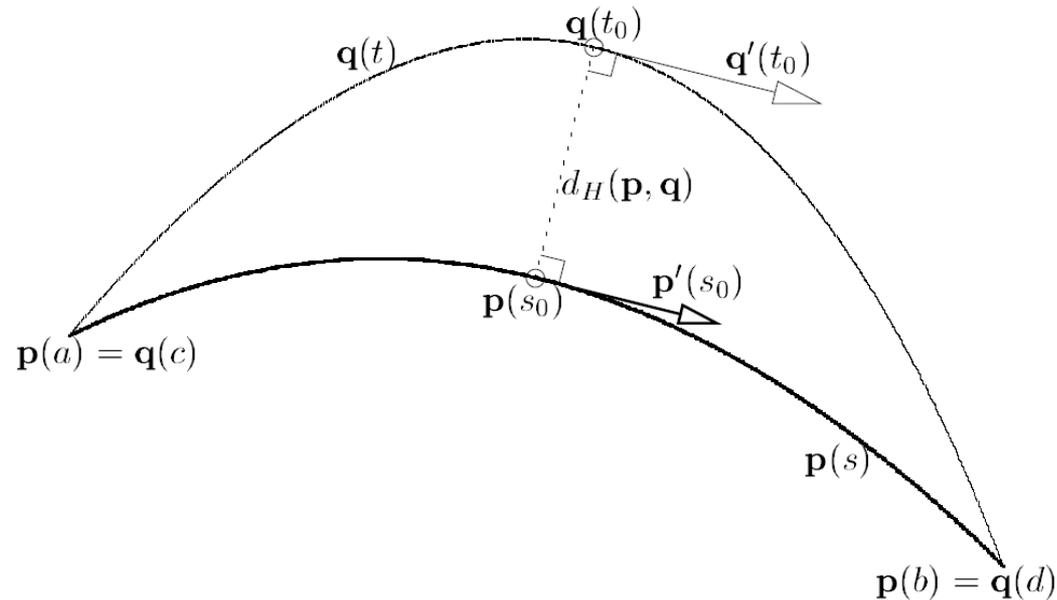
Quadratic Bézier curve (gray)

Quadratic Bézier-like curve approximation by circular arc (orange)

# Hausdorff Distance Equation

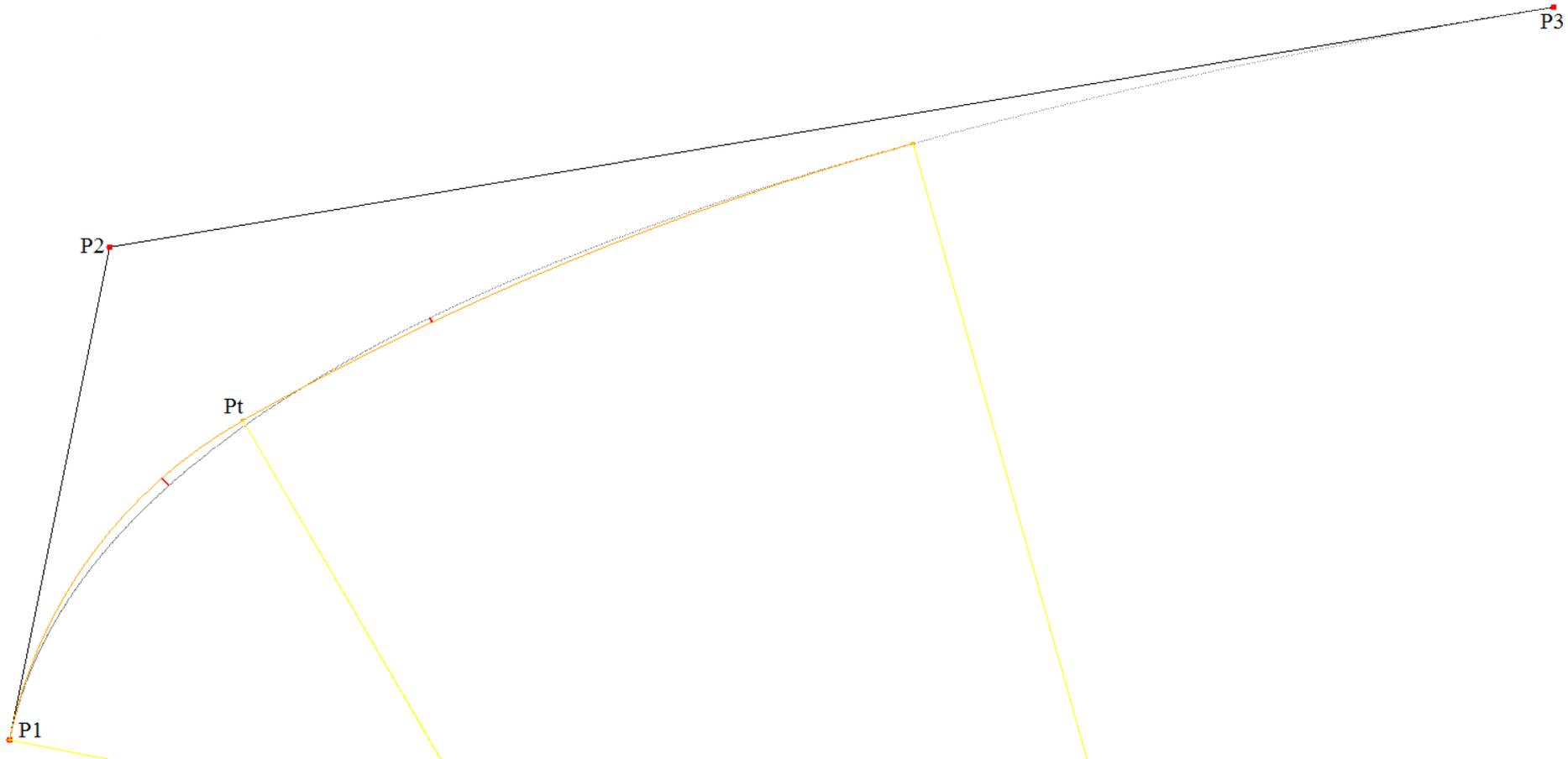
- Ahn in another work presented the exact Hausdorff Distance between the offset curve of quadratic Bézier curve and its quadratic approximation by solving a nonlinear system of two variables as follow:

$$\mathbf{p}'(s_0) \parallel \mathbf{q}'(t_0) \quad \text{and} \quad \mathbf{p}'(s_0) \perp \overrightarrow{\mathbf{p}(s_0)\mathbf{q}(t_0)}$$



# Hausdorff Distance Equation

- Every quadratic Bézier curve segment without the parabolic peak point has two Distance value which the maximum one is Hausdorff Distance.

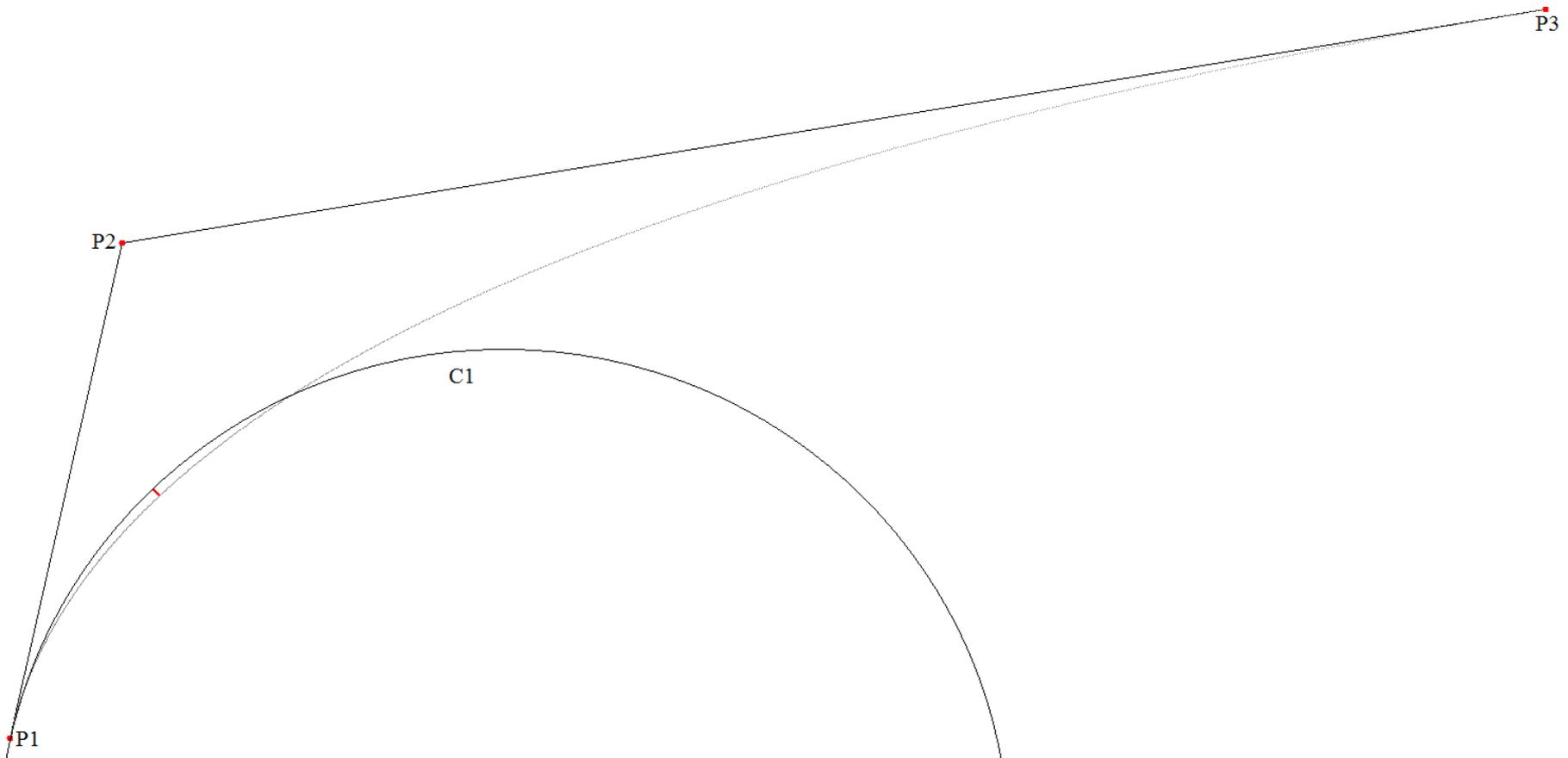


# A New Approximation Method

- The new method is not using the Bisection Algorithm and it keeps Hausdorff Distance equal to tolerance band.
- This method does not need a huge processing time to calculate Hausdorff Distance.
- If the quadratic Bézier curve contains the parabolic peak point then we have to divide it on this point and at the end we can merge two adjacent circular arc on the parabolic peak point.
- The intersection point of each biarc to cover a curve segment touches a tangent line on the curve.

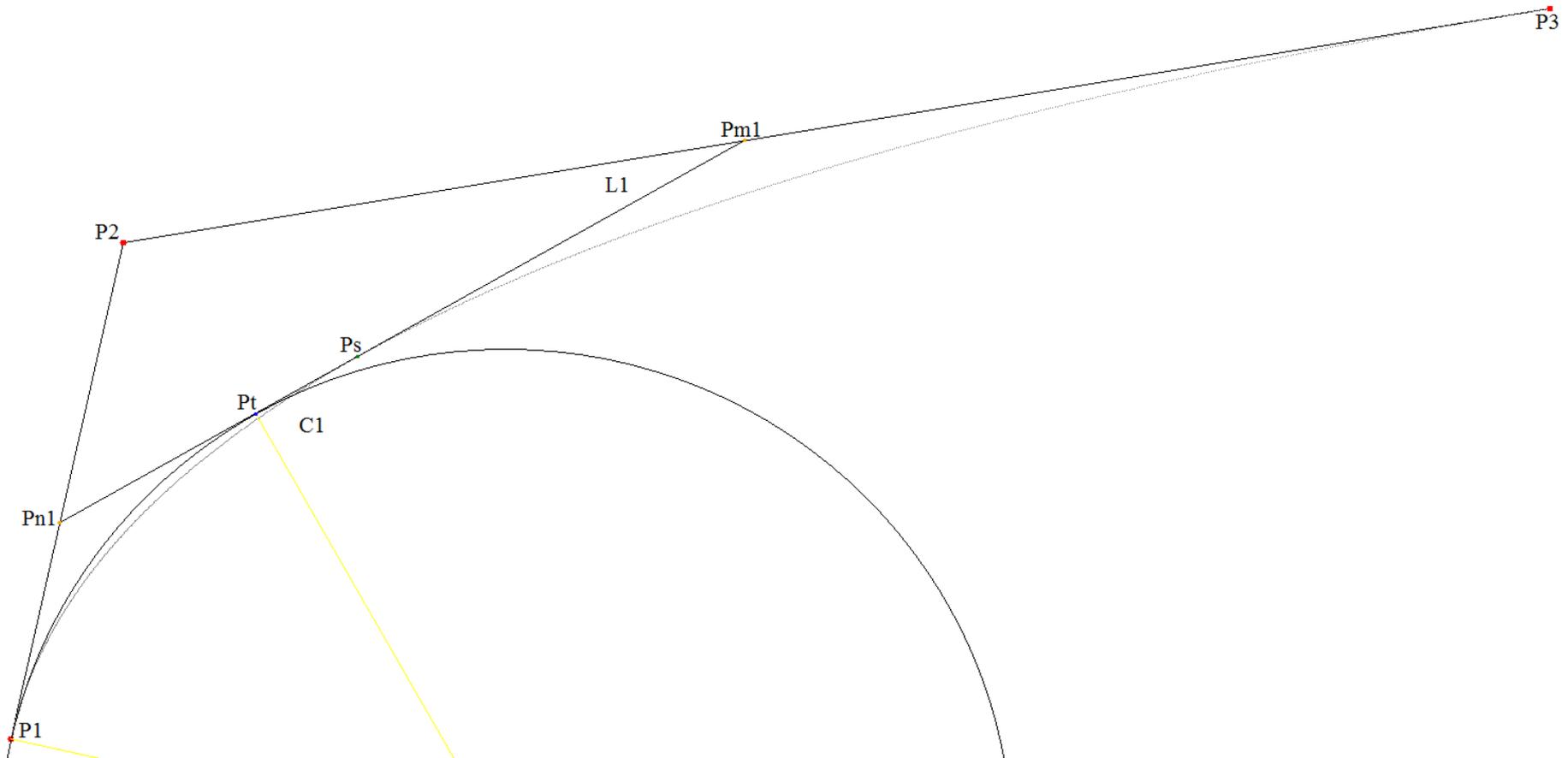
# A New Approximation Method

- In the first step we have to draw the adjacent biarc belong to the shortest control line by the Hausdorff Distance as a tolerance band.



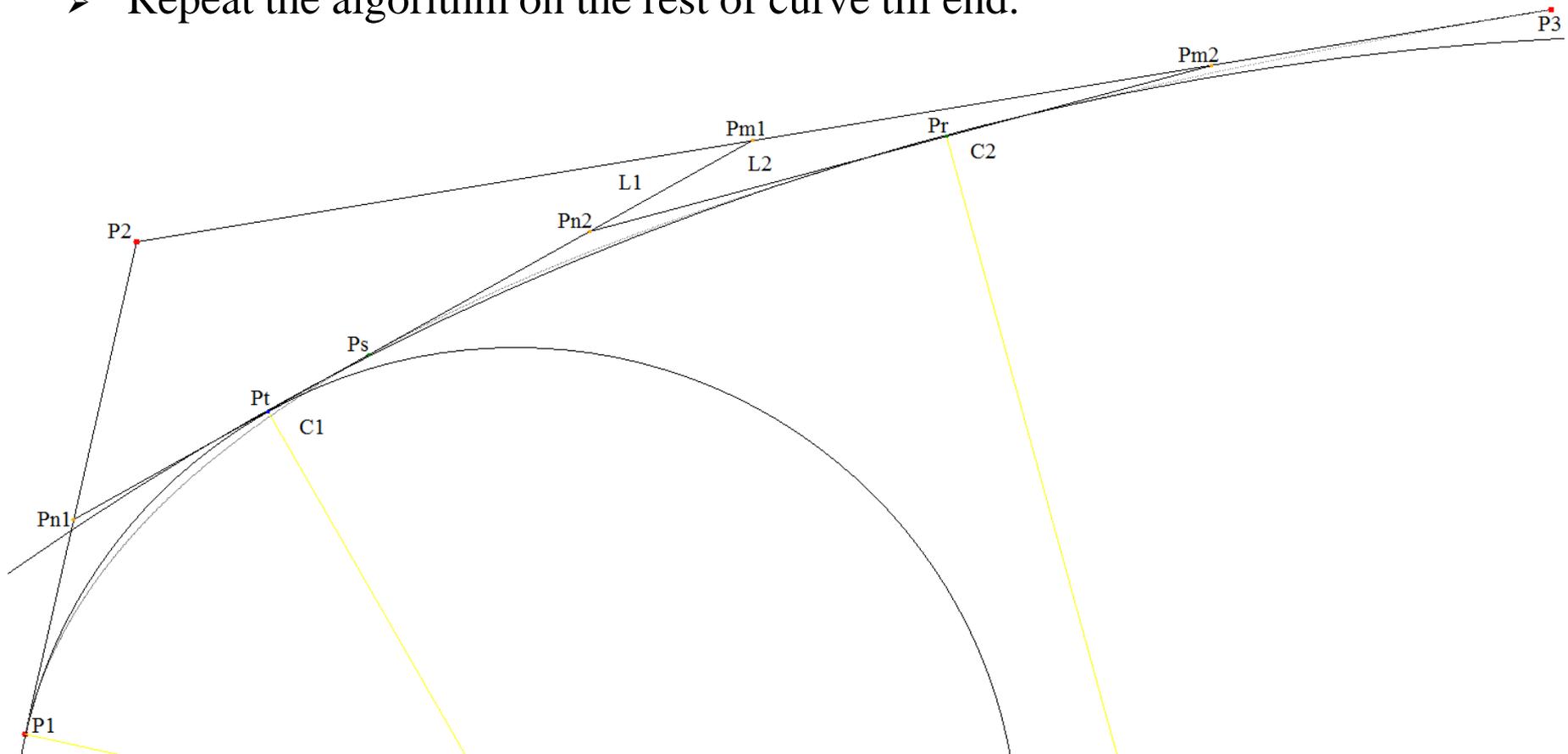
# A New Approximation Method

- In the second step we are drawing  $L1$  that is the tangent line of the corresponding biarc  $C1$  at  $P_t$  and the quadratic Bézier curve at  $P_s$ .



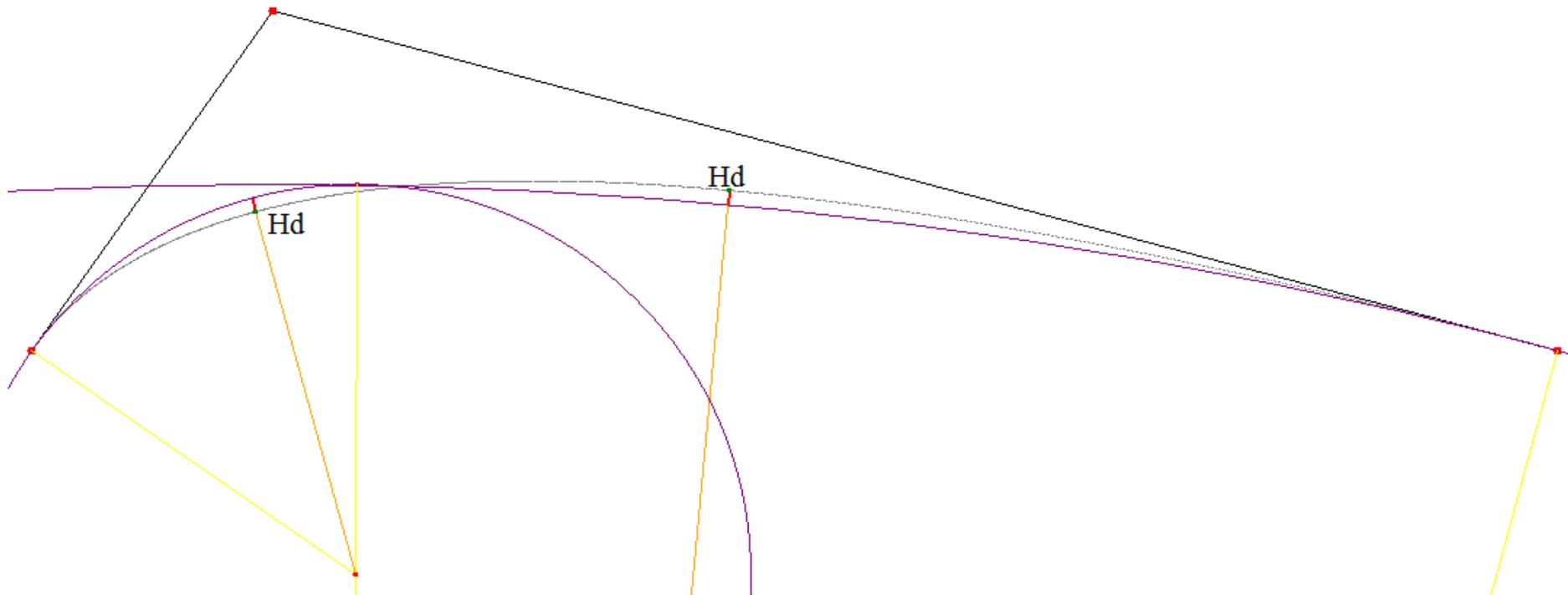
# A New Approximation Method

- At the end we find the tangent biarc C2 on the quadratic Bézier curve at  $P_r$  that is bigger than C1 and touch it at  $P_t$ .
- Repeat the algorithm on the rest of curve till end.



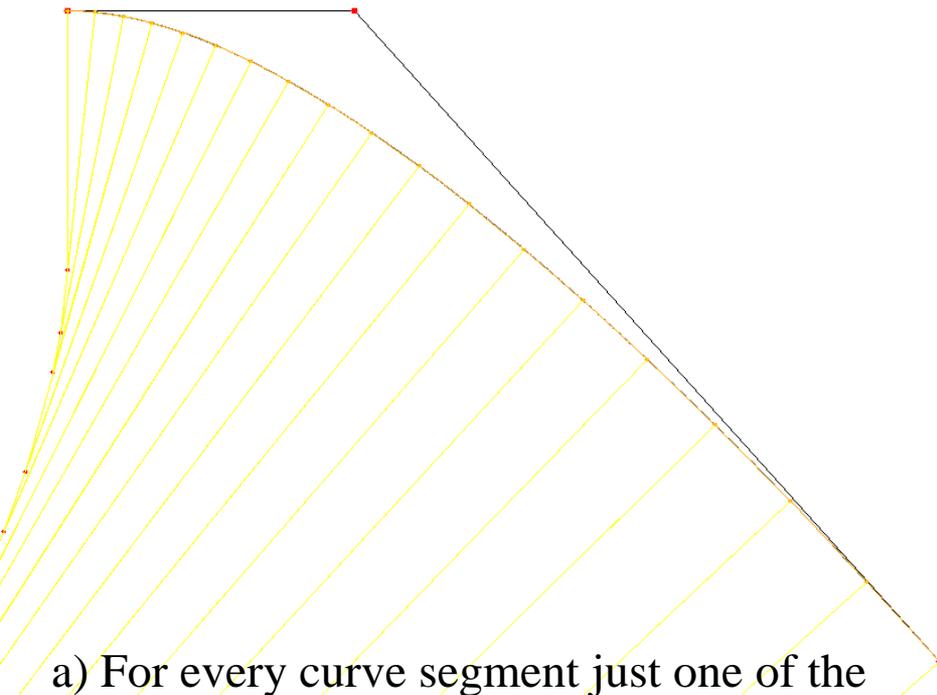
# A New Approximation Method

- The second new method is almost the same but for each quadratic Bézier curve segment, both of the biarc are belong to the Hausdorff Distance.
- Therefore they must have the same value to create the longest biarc to cover a curve segment.

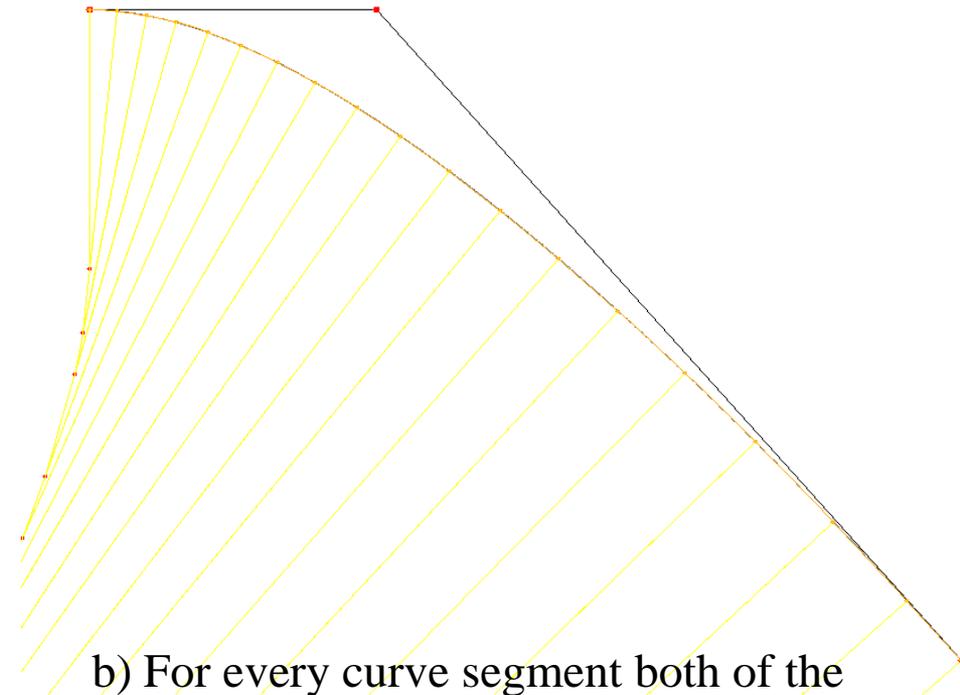


# The Algorithm Results

- The second new method can create less number of biarc.
- In the result you can see the second method is created a shorter biarc at the end.



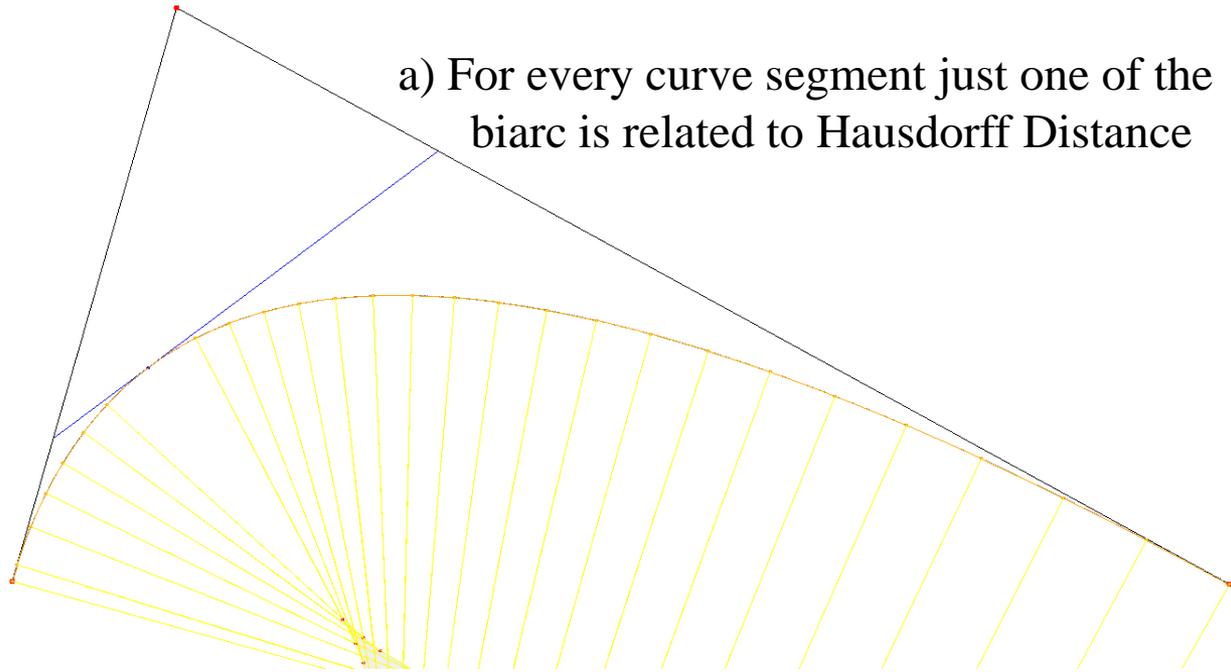
a) For every curve segment just one of the biarc is related to Hausdorff Distance



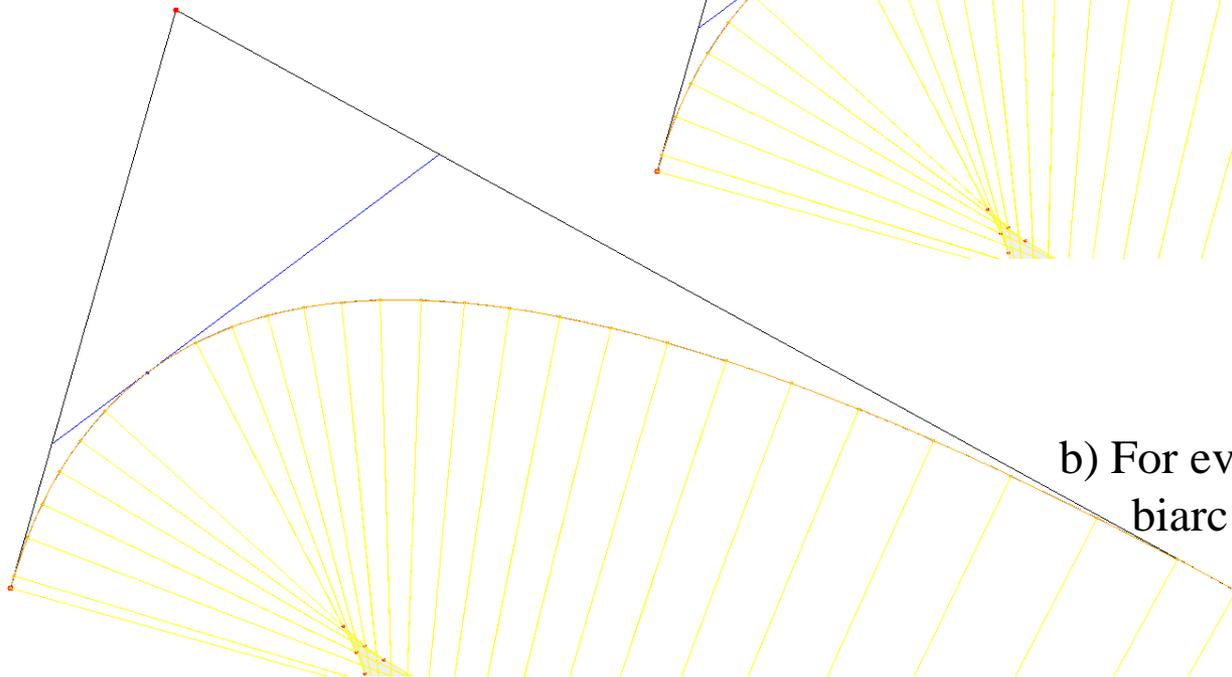
b) For every curve segment both of the biarc is related to Hausdorff Distance

# The Algorithm Results

a) For every curve segment just one of the biarc is related to Hausdorff Distance



b) For every curve segment both of the biarc is related to Hausdorff Distance





Thanks For Your Attention